MATHEMATICS OF COMPUTATION
Volume 00, Number 0, Pages 000–000
S 0025-5718(XX)0000-0

AN EFFICIENT DETERMINISTIC TEST FOR KLOOSTERMAN SUM ZEROS

OMRAN AHMADI AND ROBERT GRANGER

Abstract. We propose a simple deterministic test for deciding whether or not an element $a \in \mathbb{F}_{2^n}^{\times}$ or $\mathbb{F}_{3^n}^{\times}$ is a zero of the corresponding Kloosterman sum over these fields, and rigorously analyse its runtime. The test seems to have been overlooked in the literature. The expected cost of the test for binary fields is a single point-halving on an associated elliptic curve, while for ternary fields the expected cost is one half of a point-thirding on an associated elliptic curve. For binary fields of practical interest, this represents an $O(n)$ speedup over the previous fastest test. By repeatedly invoking the test on random elements of $\mathbb{F}_{2^n}^{\times}$ we obtain the most efficient probabilistic method to date to find non-trivial Kloosterman sum zeros. The analysis depends on the distribution of Sylow p-subgroups in the two families of associated elliptic curves, which we ascertain using a theorem due to Howe.

1. Introduction

For a finite field $\mathbb{F}_{p^n}$, the Kloosterman sum $\mathcal{K}_{p^n} : \mathbb{F}_{p^n} \to \mathbb{C}$ can be defined by

$$\mathcal{K}_{p^n}(a) = 1 + \sum_{x \in \mathbb{F}_{p^n}^{\times}} \zeta^{\mathrm{Tr}(x^{-1}+ax)},$$

where ζ is a primitive p-th root of unity and Tr denotes the absolute trace map $\mathrm{Tr} : \mathbb{F}_{p^n} \to \mathbb{F}_p$, defined by

$$\mathrm{Tr}(x) = x + x^p + x^{p^2} + \cdots + x^{p^{n-1}}.$$

Note that in some contexts the Kloosterman sum is defined to be just the summation term without the added '1' [23]. As one would expect, a Kloosterman (sum) zero is simply an element $a \in \mathbb{F}_{p^n}^{\times}$ for which $\mathcal{K}_{p^n}(a) = 0$.

Kloosterman sums have recently become the focus of much research, most notably due to their applications in cryptography and coding theory (see [6, 34] for example). In particular, zeros of $\mathcal{K}_{2^n}$ lead to bent functions from $\mathbb{F}_{2^{2n}} \to \mathbb{F}_2$ [10], and similarly zeros of $\mathcal{K}_{3^n}$ give rise to ternary bent functions [17].

It was recently shown that zeros of Kloosterman sums only exist in characteristics 2 and 3 [25], and hence these are the only cases we consider. Finding such zeros is regarded as being difficult, and recent research has tended to focus on characterising Kloosterman sums modulo small integers [7, 12–16, 28, 29, 33]. While these results are interesting in their own right, they also provide a sieve which may be used to

2010 *Mathematics Subject Classification.* 11L05, (11G20, 11T71, 11Y16).

Key words and phrases. Kloosterman sum zeros, elliptic curves, Sylow p-subgroups.

Both authors are supported by the Claude Shannon Institute, Science Foundation Ireland Grant No. 06/MI/006.

eliminate elements of a certain form prior to testing whether they are Kloosterman zeros or not, by some method.

It has long been known that Kloosterman sums over binary and ternary fields are intimately related to the group orders of members of two families of elliptic curves over these fields [23, 26, 32, 41]. In particular, for $p \in \{2, 3\}$ the Kloosterman sum $\mathcal{K}_{p^n}(a)$ is equal to one minus the trace of the Frobenius endomorphism of an associated elliptic curve $E_{p^n}(a)$. As such, one may use p-adic methods — originally due to Satoh [37] — to compute the group orders of these elliptic curves, and hence the corresponding Kloosterman sums. The best p-adic point counting method asymptotically takes $O(n^2 \log^2 n \log\log n)$ bit operations and requires $O(n^2)$ memory; see Vercauteren's thesis [42] for contributions and a comprehensive survey.

Rather than count points, Lisoněk has suggested that if instead one only wants to check whether a given element is a zero, one can do so by testing whether a random point of $E_{p^n}(a)$ has order p^n, via point multiplication [28]. Asymptotically, this has a similar bit complexity to the point counting approach, requires less memory, but is randomised. For fields of practical interest, it is reported that this approach is superior to point counting [28, §3], and using this method Lisoněk was able to find a zero of $\mathcal{K}_{2^n}$ for $n \leq 64$ and $\mathcal{K}_{3^n}$ for $n \leq 34$, in a matter of days.

In this paper we take the elliptic curve connection to a logical conclusion, in terms of proving divisibility results of Kloosterman sums by powers of the characteristic. In particular we give an efficient deterministic algorithm to compute the Sylow 2- and 3-subgroups of the associated elliptic curves in characteristics 2 and 3 respectively, along with a generator (these subgroups are cyclic in the cases considered). Moreover, the average case runtimes of the two algorithms are rigorously analysed. For binary fields of practical interest, the test gives an $O(n)$ speedup over the point multiplication test.

Finding a single Kloosterman zero — which is often all that is needed in applications — is then a matter of testing random field elements until one is found, the success probability of which crucially depends on the number of Kloosterman zeros, see [23] and §6.3. Our runtime analysis provides a non-trivial upper bound on this number, and consequently finding a Kloosterman zero with this approach still requires time exponential in the size of the field. We note that should one want to find *all* Kloosterman zeros over $\mathbb{F}_{2^n}$, rather than just one, then one can use the fast Walsh-Hadamard transform (see [2] for an overview), which requires $O(2^n \cdot n^2)$ bit operations and $O(2^n \cdot n)$ space.

The sequel is organised as follows. In §2 we detail the basic connection between Kloosterman sums and two families of elliptic curves. In §3 we present the main idea behind our algorithm, while §4 and §5 explore its specialisation to binary and ternary fields respectively. In §6 we present data on the runtime of the two algorithms, provide a heuristic analysis which attempts to explain the data, and give an exact formula for the average case runtime. In §7 we rigorously prove the expected runtime, while in §8 we assess the practical efficiency of the tests. We finally make some concluding remarks in §9.

2. Connection with elliptic curves

Our observations stem from the following three simple lemmas, which connect Kloosterman sums over $\mathbb{F}_{2^n}$ and $\mathbb{F}_{3^n}$ with the group orders of elliptic curves in two corresponding families. The first is due to Lachaud and Wolfmann [26], the second Moisio [32], while the third was proven by Lisoněk [28].

Lemma 2.1. *Let $a \in \mathbb{F}_{2^n}^{\times}$ and define the elliptic curve $E_{2^n}(a)$ over $\mathbb{F}_{2^n}$ by*

$$E_{2^n}(a) : y^2 + xy = x^3 + a.$$

Then $\#E_{2^n}(a) = 2^n + \mathcal{K}_{2^n}(a)$.

Lemma 2.2. *Let $a \in \mathbb{F}_{3^n}^{\times}$ and define the elliptic curve $E_{3^n}(a)$ over $\mathbb{F}_{3^n}$ by*

$$E_{3^n}(a) : y^2 = x^3 + x^2 - a.$$

Then $\#E_{3^n}(a) = 3^n + \mathcal{K}_{3^n}(a)$.

Lemma 2.3. *Let $p \in \{2, 3\}$, let $a \in \mathbb{F}_{p^n}^{\times}$, and let $1 \leq h \leq n$. Then $p^h \mid \mathcal{K}_{p^n}(a)$ if and only if there exists a point of order p^h on $E_{p^n}(a)$.*

Lemma 2.3 is a simple consequence of the structure theorem for elliptic curves over finite fields. Note that for $p \in \{2, 3\}$, by Lemmas 2.1 and 2.2 we have $\mathcal{K}_{p^n}(a) = 0$ if and only if $E_{p^n}(a)$ has order p^n. By Lemma 2.3, this is equivalent to $E_{p^n}(a)$ having a point of order p^n, and hence finding a point of order p^n proves that $\mathcal{K}_{p^n}(a) = 0$, since p^n is the only element divisible by p^n in the Hasse interval. For the remainder of the paper, when we refer to a prime p we implicitly presume $p \in \{2, 3\}$.

3. Determining the Sylow p-subgroup of $E_{p^n}(a)$

It is easy to show that $\mathcal{K}_{2^n}(a) \equiv 0 \pmod 4$ and $\mathcal{K}_{3^n}(a) \equiv 0 \pmod 3$ for all $a \in \mathbb{F}_{2^n}^{\times}$ and $\mathbb{F}_{3^n}^{\times}$ respectively. One way to see this is to observe that $E_{2^n}(a)$ possesses a point of order 4 (see §4) and $E_{3^n}(a)$ possesses a point of order 3 (see §5), and hence by Lagrange's theorem, $4 \mid \#E_{2^n}(a)$ and $3 \mid \#E_{3^n}(a)$.

For an integer x, let $\mathrm{ord}_p(x)$ be the exponent of the maximum power of p that divides x. For $a \in \mathbb{F}_{p^n}^{\times}$, let $h = \mathrm{ord}_p(\#E_{p^n}(a))$. By Lemma 2.3 the Sylow p-subgroup $S_p(E_{p^n}(a))$ is cyclic of order p^h, and hence has $(p-1)p^{h-1}$ generators. Multiplying these by p results in the $(p-1)p^{h-2}$ generators of the order p^{h-1} subgroup. Continuing this multiplication by p process, after $h-1$ steps one arrives at the p-torsion subgroup $E_{p^n}(a)[p]$, consisting of $p-1$ order-p points and the identity element $\mathcal{O}$. These considerations reveal the structure of the p-power torsion subgroups $E_{p^n}(a)[p^k]$ for $1 \leq k \leq h$, which one may view as a tree, with $\mathcal{O}$ as the root node. The root has $p-1$ children which are the non-identity points in $E_{p^n}(a)[p]$. If $h > 1$ each of these $p-1$ nodes has p children: the elements of $E_{p^n}(a)[p^2] \setminus E_{p^n}(a)[p]$. For $1 < k < h$, each of the $(p-1)p^{k-1}$ depth-k nodes have p children, while at depth h we have $(p-1)p^{h-1}$ leaf nodes.

Using a division polynomial approach Lisoněk was able to prove a necessary condition on $a \in \mathbb{F}_{2^n}^{\times}$ such that $\mathcal{K}_{2^n}(a)$ is divisible by 16, and likewise a necessary condition on $a \in \mathbb{F}_{3^n}^{\times}$ such that $\mathcal{K}_{3^n}(a)$ is divisible by 9. While necessary conditions for the divisibility of $\mathcal{K}_{2^n}(a)$ by 2^k have since been derived for $k \leq 8$ [13], and for the divisibility of $\mathcal{K}_{3^n}(a)$ by 3^k for $k \leq 3$ [16], these use p-adic methods; the division polynomial approach seemingly being too cumbersome to progress any further.

However, the process outlined above — taking a generator of $S_p(E_{p^n}(a))$ and multiplying by p repeatedly until the non-identity elements of the p-torsion are obtained — can be reversed, easily and efficiently, using point-halving in even characteristic, and point-thirding in characteristic three, as we demonstrate in the ensuing two sections. Furthermore, due to the cyclic structure of $S_p(E_{p^n}(a))$, at each depth, either all points are divisible by p, or none are. This means one can determine the height of the tree by using a depth-first search, without any backtracking; in particular, when a point P at a given depth can not be halved or thirded, this depth is $\log_p(|S_p(E_{p^n}(a))|)$, and P is a generator. Furthermore, one can do this without ever computing the group order of the curve.

This process has been considered previously by Miret *et al.*, for determining the Sylow 2-subgroup of elliptic curves over arbitrary finite fields of characteristic > 2 [30]; for $p = 2$ the algorithm follows easily from the above considerations and point-halving, which is well studied in cryptographic circles [1,24,38], and is known to be more than twice as fast as point-doubling in some cases [11]. For primes $l > 2$, Miret *et al.* also addressed how to compute the Sylow l-subgroup of elliptic curves over arbitrary finite fields provided that l was not the characteristic of the field [31]. Therefore we address here the case $l = p = 3$, for the family of curves $E_{3^n}(a)$.

We summarise this process in Algorithm 1. Regarding notation, we say that a point P is p-divisible if there exists a point Q such that $[p]Q = P$, and write $Q = [1/p]P$.

Algorithm 1: **DETERMINE** $S_p(E_{p^n}(a))$

```
INPUT:  $a \in \mathbb{F}_{p^n}^{\times}$,  $P \in E_{p^n}(a)[p] \setminus \{\mathcal{O}\}$
OUTPUT: $(h, P_h)$ where $h = \mathrm{ord}_p(\#E_{p^n}(a))$ and $\langle P_h \rangle = S_p(E_{p^n}(a))$
counter $\leftarrow 1$;
While $P$ is $p$-divisible do:
    P := [1/p]P;
    counter++;
Return (counter, $P$)
```

Observe that Algorithm 1 is deterministic, provided that a deterministic method of dividing a p-divisible point by p is fixed once and for all, which we do for $p = 2$ and $p = 3$ in §4 and §5 respectively. For a given field extension under consideration, choosing an appropriate field representation and basis can also be performed deterministically, via sequential search, however we consider this to be part of the setup phase and do not incorporate setup costs when assessing the runtime of Algorithm 1.

4. Binary fields

We now work out the details of Algorithm 1 for the family of curves $E_{2^n}(a)$. For a fixed n, given a point $P = (x, y) \in E_{2^n}(a)$, $[2]P = (\xi, \eta)$ is given by the formula:

$$\begin{aligned} \lambda &= x + y/x, \\ \xi &= \lambda^2 + \lambda, \\ \eta &= x^2 + \xi(\lambda + 1). \end{aligned} \tag{4.1}$$

To halve a point, one needs to reverse this process, i.e., given $Q = (\xi, \eta)$, find (if possible) a $P = (x, y) \in E_{2^n}(a)$ such that $[2]P = Q$. To do so, one first needs to solve (4.1) for λ, which has a solution in $\mathbb{F}_{2^n}$ if and only if $\mathrm{Tr}(\xi) = 0$, since the trace of the right-hand side is zero for every $\lambda \in \mathbb{F}_{2^n}$, and one can provide an explicit solution in this case, as detailed in §4.1. Observe that if λ is a solution to (4.1) then so is $\lambda + 1$. Assuming λ has been computed, one then has

$$\begin{aligned} x &= (\eta + \xi(\lambda + 1))^{1/2}, \\ y &= x(x + \lambda), \end{aligned}$$

which for the two choices of λ gives both points whose duplication is $Q = (\xi, \eta)$.

Aside from the cost of computing λ, the computation of $P = (x, y)$ as above requires two field multiplications. As detailed in Algorithm 2, this can be reduced to just one by using the so-called λ-representation of a point [24,38], where an affine point $Q = (\xi, \eta)$ is instead represented by (ξ, λ_Q), with

$$\lambda_Q = \xi + \frac{\eta}{\xi}.$$

In affine coordinates, there is a unique 2-torsion point $(0, a^{1/2})$, which halves to the two order 4 points $P_4^+ = (a^{1/4}, a^{1/2})$, $P_4^- = (a^{1/4}, a^{1/2} + a^{1/4})$. The corresponding λ-representations of each of these are $(a^{1/4}, 0)$ and $(a^{1/4}, 1)$ respectively. For simplicity, we choose to use the former as the starting point in Algorithm 2.

ALGORITHM 2: **DETERMINE** $S_2(E_{2^n}(a))$

```
INPUT:  $a \in \mathbb{F}_{2^n}^\times$, $(x = a^{1/4}, \lambda = 0)$
OUTPUT: $(h, P_h)$ where $h = \mathrm{ord}_2(\#E_{2^n}(a))$ and $\langle P_h \rangle = S_2(E_{2^n}(a))$
counter $\leftarrow 2$;
While Tr$(x) = 0$ do:
    Solve $\widehat{\lambda}^2 + \widehat{\lambda} + x = 0$;
    $t \leftarrow x(x + \lambda + \widehat{\lambda})$;
    $x \leftarrow \sqrt{t}$;
    $\lambda \leftarrow \widehat{\lambda} + 1$;
    counter++;
Return (counter, $P = (x, x(x + \lambda))$)
```

Observe that if the x-coordinate $a^{1/4}$ of $P_4^{\pm}$ satisfies $\mathrm{Tr}(a^{1/4}) = \mathrm{Tr}(a) = 0$, then there exist four points of order 8, and hence $8 \mid \mathcal{K}_{2^n}(a)$, which was first observed by van der Geer and van der Vlugt [41], and later by several others [8, 18, 28].

4.1. **Solving $\widehat{\lambda}^2 + \widehat{\lambda} + x = 0$.** For odd n, let $\widehat{\lambda}$ be given by the following function, which is known as the *half trace*:

$$\widehat{\lambda}(x) = \sum_{i=0}^{(n-1)/2} x^{2^{2i}}. \tag{4.2}$$

One can easily verify that this $\widehat{\lambda}$ satisfies the stated equation. When n is even, the half trace approach will not work, essentially because $\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(1) = 0$. Hence fix an element $\delta \in \mathbb{F}_{2^n}$ with $\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(\delta) = 1$. Such a δ can be found during the setup phase via the sequential search of the trace of the polynomial basis elements, or by using the methods of [1]. A solution to equation (4.1) is then given by [3, Chapter II]:

$$\widehat{\lambda}(x) = \sum_{i=0}^{n-2}\left(\sum_{j=i+1}^{n-1} \delta^{2^j}\right)x^{2^i}, \tag{4.3}$$

as may be verified. Note that for odd n, $\delta = 1$ suffices and so (4.3) simplifies to (4.2). The inner sums of equation (4.3) can be precomputed, and for a general $\delta \in \mathbb{F}_{2^n}$ the computation of $\widehat{\lambda}(x)$ would require $n-1$ multiplications in $\mathbb{F}_{2^n}$, which together with the multiplication coming from `line 4` of Algorithm 2, gives a total of n full $\mathbb{F}_{2^n}$-multiplications.

However, should $\mathbb{F}_{2^n}$ contain a subfield of odd index, then one can reduce this cost as follows. Let $n = 2^m n'$ with $m \geq 1$ and n' odd. Constructing $\mathbb{F}_{2^n}$ as a degree n' extension of $\mathbb{F}_{2^{2^m}}$, fix a $\delta \in \mathbb{F}_{2^{2^m}}$ with $\mathrm{Tr}_{\mathbb{F}_{2^{2^m}}/\mathbb{F}_2}(\delta) = 1$. Then

$$\mathrm{Tr}_{\mathbb{F}_{2^{2^m \cdot n'}}/\mathbb{F}_2}(\delta) = n' \cdot \mathrm{Tr}_{\mathbb{F}_{2^{2^m}}/\mathbb{F}_2}(\delta) = 1.$$

Hence this δ can be used in (4.3). As $\delta^{2^{2^m}} = \delta$, upon expanding (4.3) in terms of $\{\delta^{2^0}, \delta^{2^1}, \ldots, \delta^{2^{2^m-1}}\}$, we see that at most 2^m multiplications of elements of $\mathbb{F}_{2^{2^m}}$ by elements of $\mathbb{F}_{2^n}$ are required. So the smaller the largest power of 2 dividing n is, the faster one can compute $\widehat{\lambda}(x)$.

However, since the expressions for $\widehat{\lambda}(x)$ in (4.2) and (4.3) are linear maps, in practice it is far more efficient for both odd and even n to precompute and store $\{\widehat{\lambda}(t^i)\}_{i=0,\ldots,n-1}$ during setup, where $\mathbb{F}_{2^n} = \mathbb{F}_2(t)$ and $x = \sum_{i=0}^{n-1} x_i t^i$. One then has

$$\widehat{\lambda}(x) = \sum_{i=0}^{n-1} x_i \widehat{\lambda}(t^i).$$

On average just $n/2$ additions in $\mathbb{F}_{2^n}$ are required for each point-halving. Both the storage required and execution time can be further reduced [11]. We defer consideration of the practical efficiency of Algorithm 2 until §8.2.

5. Ternary fields

Let $Q = (\xi, \eta) \in E_{3^n}(a)$. To find $P = (x, y)$ such that $[3]P = Q$, when possible, we do the following. As in [31, §4], we have

$$x([3]P) = x(P) - \frac{\Psi_2(x,y)\Psi_4(x,y)}{\Psi_3^2(x,y)},$$

or

$$(x - \xi)\Psi_3^2(x,y) - \Psi_2(x,y)\Psi_4(x,y) = 0,$$

where Ψ_l is the l-th division polynomial. Working modulo the equation of $E_{3^n}(a)$, this becomes

$$x^9 - \xi x^6 + a(1-\xi)x^3 - a^2(a+\xi) = 0,$$

whereupon substituting $X = x^3$ gives

$$f(X) = X^3 - \xi X^2 + a(1-\xi)X - a^2(a+\xi) = 0. \tag{5.1}$$

To solve (5.1), we make the transformation

$$g(X) = X^3 f\left(\frac{1}{X} - \frac{a(1-\xi)}{\xi}\right) = \frac{a^2\eta^2}{\xi^3}X^3 - \xi X + 1.$$

Hence we must solve

$$X^3 - \frac{\xi^4}{a^2\eta^2}X + \frac{\xi^3}{a^2\eta^2} = 0.$$

Writing $X = \frac{\xi^2}{a\eta}\widehat{X}$ this becomes

$$\widehat{X}^3 - \widehat{X} + \frac{a\eta}{\xi^3} = 0. \tag{5.2}$$

Our thirding condition is then simply $\mathrm{Tr}(a\eta/\xi^3) = 0$, since as in the binary case, for every element $\widehat{X} \in \mathbb{F}_{3^n}$ we have $\mathrm{Tr}(\widehat{X}^3 - \widehat{X}) = 0$, and if so then one can provide an explicit solution, as detailed in §5.1. Observe that if $\widehat{X}$ is a solution to (5.2) then so is $\widehat{X} \pm 1$. Unrolling the transformations leads to the following algorithm, with input the 3-torsion point $P_3 = (a^{1/3}, a^{1/3})$.

Algorithm 3: **DETERMINE** $S_3(E_{3^n}(a))$

```
INPUT:  $a \in \mathbb{F}_{3^n}^{\times}$, $(x = a^{1/3}, y = a^{1/3})$
OUTPUT: $(h, P_h)$ where $h = \mathrm{ord}_3(\#E_{3^n}(a))$ and $\langle P_h \rangle = S_3(E_{3^n}(a))$
counter $\leftarrow 1$;
While Tr$(ay/x^3) = 0$ do:
    Solve $\widehat{X}^3 - \widehat{X} + \frac{ay}{x^3} = 0$;
    $x \leftarrow \left(\frac{ay}{x^2\widehat{X}} - \frac{a(1-x)}{x}\right)^{1/3}$;
    $y \leftarrow \left(x^3 + x^2 - a\right)^{1/2}$;
    counter++;
Return (counter, $P = (x, y)$)
```

Observe that as with Algorithm 2, if the point P_3 satisfies $\mathrm{Tr}(a \cdot a^{1/3}/a) = \mathrm{Tr}(a) = 0$, then there is a point of order 9, and hence $9 \mid \mathcal{K}_{3^n}(a)$, which again was first proven in [41], and later by others [14, 28].

5.1. **Solving** $\widehat{X}^3 - \widehat{X} + \frac{ay}{x^3} = 0$. Let $\beta = \frac{ay}{x^3}$, and let $\delta \in \mathbb{F}_{3^n}$ be an element with $\mathrm{Tr}_{\mathbb{F}_{3^n}/\mathbb{F}_3}(\delta) = 1$, which can be found deterministically during the setup phase. It is then a simple matter to verify that

$$\widehat{X}(\beta) = \sum_{i=0}^{n-2}\left(\sum_{j=i+1}^{n-1}\delta^{3^j}\right)\beta^{3^i} \tag{5.3}$$

is a solution to equation (5.2).

For $n \equiv 1 \pmod 3$, one may choose $\delta = 1$ and the expression for $\widehat{X}(\beta)$ in equation (5.3) simplifies to

$$\widehat{X}(\beta) = \sum_{i=1}^{(n-1)/3}\left(\beta^{3^{3i-1}} - \beta^{3^{3i-2}}\right).$$

For $n \equiv 2 \pmod 3$, one may choose $\delta = -1$ and the expression for $\widehat{X}(\beta)$ in equation (5.3) simplifies to

$$\widehat{X}(\beta) = -\beta + \sum_{i=1}^{(n-2)/3}\left(\beta^{3^{3i-1}} - \beta^{3^{3i}}\right).$$

For $n \equiv 0 \pmod 3$, one can use the approach described in §4.1 to pick δ from the smallest subfield of $\mathbb{F}_{3^n}$ of index coprime to 3, in order to reduce the cost and the number of multiplications required to solve (5.2). As in the binary case, one can also exploit the linearity of $\widehat{X}(\beta)$ and precompute and store $\{\widehat{X}(t^i)\}_{i=0,\ldots,n-1}$ during setup, where $\mathbb{F}_{3^n} = \mathbb{F}_3(t)$ and $\beta = \sum_{i=0}^{n-1} \beta_i t^i$, in order to reduce the cost of solving (5.2) to an average of $2n/3$ additions. We defer consideration of the practical efficiency of Algorithm 3 until §8.3.

6. Heuristic analysis of the expected number of iterations

For any input $a \in \mathbb{F}_{p^n}^{\times}$, the runtime of Algorithm 1 is proportional to the number of loop iterations performed, which is precisely the height of the corresponding Sylow p-subgroup tree, $h = \log_p(|S_p(E_{p^n}(a))|)$. In this section we present experimental data for the distribution of these heights for $p \in \{2,3\}$, provide a heuristic argument to explain them, and give an exact formula for the average case runtime. Since we are interested in the average number of loop iterations[1], we consider the arithmetic mean of the heights of the Sylow p-subgroup trees, or equivalently the logarithm of the geometric mean of their orders.

6.1. **Experimental data.**

In order to gain an idea of how $\{\log_p(|S_p(E_{p^n}(a))|)\}_{a \in \mathbb{F}_{p^n}^{\times}}$ is distributed, we computed all of them for several small extensions of $\mathbb{F}_p$. Tables 1 and 2 give the results for $p = 2$ and $p = 3$ respectively.

Observe that for $p = 2$, the first two columns are simply $2^n - 1 = |\mathbb{F}_{2^n}^{\times}|$, reflecting the fact that all of the curves $\{E_{2^n}(a)\}_{a \in \mathbb{F}_{2^n}^{\times}}$ have order divisible by 4. Similarly for $p = 3$, the first column is given by $3^n - 1 = |\mathbb{F}_{3^n}^{\times}|$, reflecting the fact that all the curves $\{E_{3^n}(a)\}_{a \in \mathbb{F}_{3^n}^{\times}}$ have order divisible by 3. Furthermore, since exactly half of the elements of $\mathbb{F}_{2^n}$ have zero trace, the third column for $p = 2$ is given by $2^{n-1} - 1$. Likewise for $p = 3$, the second column is given by $3^{n-1} - 1$, since exactly one third of the elements of $\mathbb{F}_{3^n}$ have zero trace. For $p = 2$ there is an elegant result due Lisoněk and Moisio which gives a closed formula for the n-th entry of column 4 of Table 1 [29, Theorem 3.6], which includes the $a = 0$ case, namely:

$$(2^n - (-1+i)^n - (-1-i)^n)/4. \tag{6.1}$$

Beyond these already-explained columns, it appears that as one successively moves one column to the right, the number of such a decreases by an approximate factor of 2 or 3 respectively, until the number of Kloosterman zeros is reached, which by Hasse bound occurs as soon as $p^k > 1 + 2p^{n/2}$, or $k > n/2 + \log_p 2$.

6.2. **A heuristic for the expected number of iterations.**

To explain the data in Tables 1 and 2, we propose the following simple heuristic (and prove the validity of its consequences in §7):

Heuristic 6.1. *Over all $a \in \mathbb{F}_{p^n}^{\times}$, on any occurrence of* `line 2` *of the loop in Algorithms 2 and 3, regardless of the height of the tree at that point, the argument of the $\mathbb{F}_{p^n}$ trace is uniformly distributed over $\mathbb{F}_{p^n}$, and hence is zero with probability $1/p$.*

[1] The worst case being n iterations, which of course is the best case when searching for a Kloosterman zero.

Table 1. $\#\{E_{2^n}(a)\}_{a\in\mathbb{F}_{2^n}^{\times}}$ whose group order is divisible by 2^k

$n\backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1											
2	3	3											
3	7	7	3										
4	15	15	7	5									
5	31	31	15	5	5								
6	63	63	31	15	12	12							
7	127	127	63	35	14	14	14						
8	255	255	127	55	21	16	16	16					
9	511	511	255	135	63	18	18	18	18				
10	1023	1023	511	255	125	65	60	60	60	60			
11	2047	2047	1023	495	253	132	55	55	55	55	55		
12	4095	4095	2047	1055	495	252	84	72	72	72	72	72	
13	8191	8191	4095	2015	1027	481	247	52	52	52	52	52	52

Table 2. $\#\{E_{3^n}(a)\}_{a\in\mathbb{F}_{3^n}^{\times}}$ whose group order is divisible by 3^k

$n\backslash k$	1	2	3	4	5	6	7	8	9	10	11
1	2										
2	8	2									
3	26	8	3								
4	80	26	4	4							
5	242	80	35	15	15						
6	728	242	83	24	24	24					
7	2186	728	266	77	21	21	21				
8	6560	2186	692	252	48	48	48	48			
9	19682	6560	2168	741	270	108	108	108	108		
10	59048	19682	6605	2065	575	100	100	100	100	100	
11	177146	59048	19547	6369	2596	924	264	264	264	264	264

While this assumption is clearly false at depths $> n/2 + \log_p 2$, the data in Tables 1 and 2 does support it (up to relatively small error terms). In order to calculate the expected value of $\log_p(|S_p(E_{p^n}(a))|)$, we think of Algorithms 2 and 3 as running on all $p^n - 1$ elements of $\mathbb{F}_{p^n}^{\times}$ in parallel; we then sum the number of elements which survive the first loop, then the second loop and the third loop etc., and divide this sum by $p^n - 1$ to give the average. We now explore the consequences of Heuristic 6.1, treating the two characteristics in turn.

For Algorithm 2, on the first occurrence of `line 2`, $2^{n-1} - 1$ elements of $\mathbb{F}_{2^n}^{\times}$ have zero trace and hence $2^{n-1} - 1$ elements require an initial loop iteration. On the second occurrence of `line 2`, by Heuristic 6.1, approximately $2^{n-1}/2 = 2^{n-2}$ of the inputs have zero trace and so this number of loop iterations are required. Continuing in this manner and summing over all loop iterations at each depth, one obtains a total of

$$2^{n-1} + 2^{n-1} + \cdots + 2 + 1 \approx 2^n,$$

for the number of iterations that need to be performed for all $a \in \mathbb{F}_{2^n}^{\times}$. Thus on average this is approximately one loop iteration per initial element a. Incorporating the divisibility by 4 of all curve orders, the expected value as $n \to \infty$ of $\log_2(|S_2(E_{2^n}(a))|)$ is 3, and hence the geometric mean of $\{|S_2(E_{2^n}(a))|\}_{a \in \mathbb{F}_{2^n}^{\times}}$ as $n \to \infty$ is $2^3 = 8$.

For Algorithm 3, applying Heuristic 6.1 and the same reasoning as before, the total number of iterations required for all $a \in \mathbb{F}_{3^n}^{\times}$ is

$$3^{n-1} + 3^{n-2} + \cdots + 3 + 1 \approx 3^n/2.$$

Thus on average this is approximately 1/2 an iteration per initial element a, and incorporating the divisibility by 3 of all curve orders, the expected value as $n \to \infty$ of $\log_3(|S_3(E_{3^n}(a))|)$ is 3/2, and hence the geometric mean of $\{|S_3(E_{3^n}(a))|\}_{a \in \mathbb{F}_{3^n}^{\times}}$ as $n \to \infty$ is $3^{3/2} = 3\sqrt{3}$.

6.3. **Exact formula for the average height of Sylow p-subgroup trees.** Let $p^n + t$ be an integer in the Hasse interval $I_{p^n} = [p^n + 1 - 2p^{n/2}, p^n + 1 + 2p^{n/2}]$, which is assumed to be divisible by 4 if $p = 2$ and divisible by 3 if $p = 3$. Let $N(t)$ be the number of solutions in $\mathbb{F}_{p^n}^{\times}$ to $\mathcal{K}_{p^n}(a) = t$. The sum of the heights of the Sylow p-subgroup trees, over all $a \in \mathbb{F}_{p^n}^{\times}$, is

$$T_{p^n} = \sum_{(p^n+t) \in I_{p^n}} N(t) \cdot \mathrm{ord}_p(p^n + t), \tag{6.2}$$

and thus the expected value of $\log_p(|S_p(E_{p^n}(a))|)$ is $T_{p^n}/(p^n - 1)$. The crucial function $N(t)$ in (6.2) has been evaluated by Katz and Livné in terms of class numbers [23]. In particular, let $\alpha = (t - 1 + \sqrt{(t-1)^2 - 4p^n})/2$ for t as above. Then

$$N(t) = \sum_{\text{orders } \mathcal{O}} h(\mathcal{O}),$$

where the sum is over all orders $\mathcal{O} \subset \mathbb{Q}(\alpha)$ which contain $\mathbb{Z}[\alpha]$. It seems difficult to prove Heuristic 6.1 or our implied estimates for T_{p^n} using the Katz-Livné result directly. However, using a natural decomposition of T_{p^n} and a theorem due to Howe [20], in the following section we show that the consequences of Heuristic 6.1 as derived in §6.2 are correct.

7. Main result

We now present and prove our main result, which states that the expected value of $\{\log_p(|S_p(E_{p^n}(a))|)\}_{a \in \mathbb{F}_{p^n}^{\times}}$ is precisely as we derived heuristically in §6.2. To facilitate our analysis, for $1 \leq k \leq n$, we partition T_{p^n} into the counting functions

$$T_{p^n}(k) = \sum_{(p^n+t) \in I_{p^n}, p^k | (p^n+t)} N(t), \tag{7.1}$$

so that by (6.2) we have

$$T_{p^n} = \sum_{k=1}^{n} T_{p^n}(k). \tag{7.2}$$

Indeed, the integers $T_{p^n}(k)$ are simply the (n,k)-th entries of Tables 1 and 2 for $p = 2$ and 3 respectively, and thus T_{p^n} is the sum of the n-th row terms. Hence we already have $T_{2^n}(1) = T_{2^n}(2) = 2^n - 1$, $T_{2^n}(3) = 2^{n-1} - 1$ and $T_{2^n}(4) = (2^n - (-1 + i)^n - (-1 - i)^n)/4$ by (6.1), and similarly $T_{3^n}(1) = 3^n - 1$ and $T_{3^n}(2) = 3^{n-1} - 1$.

7.1. **Estimating** $T_{p^n}(k)$**.** For $k \geq 2$, let $\mathcal{T}_{2^n}(k)$ be the set of $\mathbb{F}_{2^n}$-isomorphism classes of elliptic curves $E/\mathbb{F}_{2^n}$ such that $\#E(\mathbb{F}_{2^n}) \equiv 0 \pmod{2^k}$. Similarly for $k \geq 1$, let $\mathcal{T}_{3^n}(k)$ be the set of $\mathbb{F}_{3^n}$-isomorphism classes of elliptic curves $E/\mathbb{F}_{3^n}$ such that $\#E(\mathbb{F}_{3^n}) \equiv 0 \pmod{3^k}$. Observe that the elliptic curves $E_{2^n}(a)$ and $E_{3^n}(a)$ both have j-invariant $1/a$ [40, Appendix A], and hence cover all the $\overline{\mathbb{F}}_{2^n}$- and $\overline{\mathbb{F}}_{3^n}$-isomorphism classes of elliptic curves over $\mathbb{F}_{2^n}$ and $\mathbb{F}_{3^n}$ respectively, except for $j = 0$. We have the following lemma.

Lemma 7.1. *[5, Lemma 6] Let $E/\mathbb{F}_q$ be an elliptic curve and let $[E]_{\mathbb{F}_q}$ be the set of $\mathbb{F}_q$-isomorphism classes of elliptic curves that are $\overline{\mathbb{F}}_q$-isomorphic to E. Then for $j \neq 0, 1728$ we have $\#[E]_{\mathbb{F}_q} = 2$, and $[E]_{\mathbb{F}_q}$ consists of the $\mathbb{F}_q$-isomorphism class of E and the $\mathbb{F}_q$-isomorphism class of its quadratic twist E^t.*

Let $\#E_{2^n}(a) = 2^n + 1 - t_a$, with t_a the trace of Frobenius. Since $j \neq 0$, by Lemma 7.1 the only other $\mathbb{F}_{2^n}$-isomorphism class with j-invariant $1/a$ is that of the quadratic twist $E^t_{2^n}(a)$, which has order $2^n + 1 + t_a$. Since $t_a \equiv 1 \pmod 4$, we have $\#E^t_{2^n}(a) \equiv 2 \pmod 4$ and hence none of the $\mathbb{F}_{2^n}$-isomorphism classes of the quadratic twists of $E_{2^n}(a)$ for $a \in \mathbb{F}_{2^n}^\times$ are in $\mathcal{T}_{2^n}(k)$, for $k \geq 2$. By an analogous argument, only the $\mathbb{F}_{3^n}$-isomorphism classes of $E_{3^n}(a)$ for $a \in \mathbb{F}_{3^n}^\times$ are in $\mathcal{T}_{3^n}(k)$, for $k \geq 1$. Furthermore, all curves $E/\mathbb{F}_{2^n}$ and $E/\mathbb{F}_{3^n}$ with $j = 0$ are supersingular [43, §3.1], and therefore have group orders $\equiv 1 \pmod 4$ and $\equiv 1 \pmod 3$ respectively. Hence no $\mathbb{F}_{p^n}$-isomorphism classes of curves with $j = 0$ are in $\mathcal{T}_{p^n}(k)$ for $p \in \{2, 3\}$. As a result, for $2 \leq k \leq n$ we have

$$|\mathcal{T}_{2^n}(k)| = T_{2^n}(k), \tag{7.3}$$

and similarly, for $1 \leq k \leq n$ we have

$$|\mathcal{T}_{3^n}(k)| = T_{3^n}(k).$$

Therefore in both cases, a good estimate for $|\mathcal{T}_{p^n}(k)|$ is all we need to estimate $T_{p^n}(k)$. The cardinality of $\mathcal{T}_{p^n}(k)$ is naturally related to the study of modular curves; in particular, considering the number of $\mathbb{F}_{p^n}$-rational points on the Igusa curve of level p^k allows one to prove Theorem 7.3 below [21, 36]. However, for simplicity (and generality) we use a result due to Howe on the group orders of elliptic curves over finite fields [20]. Consider the set

$$V(\mathbb{F}_q; N) = \{E/\mathbb{F}_q : N \mid \#E(\mathbb{F}_q)\}/\cong_{\mathbb{F}_q}$$

of equivalence classes of $\mathbb{F}_q$-isomorphic curves whose group orders are divisible by N. Following Lenstra [27], rather than estimate $V(\mathbb{F}_q; N)$ directly, Howe considers the weighted cardinality of $V(\mathbb{F}_q; N)$, where for a set S of $\mathbb{F}_q$-isomorphism classes of elliptic curves over $\mathbb{F}_q$, this is defined to be:

$$\#'S = \sum_{[E] \in S} \frac{1}{\#\mathrm{Aut}_{\mathbb{F}_q}(E)}.$$

For $j \neq 0$ we have $\#\mathrm{Aut}_{\overline{\mathbb{F}}_q}(E) = 2$ [40, §III.10] and since $\{\pm 1\} \subset \mathrm{Aut}_{\mathbb{F}_q}(E)$ we have $\#\mathrm{Aut}_{\mathbb{F}_q}(E) = 2$ also. Therefore, by the above discussion, for $p = 2, k \geq 2$ and $p = 3, k \geq 1$ we have

$$|\mathcal{T}_{p^n}(k)| = 2 \cdot \#'V(\mathbb{F}_{p^n}; p^k), \tag{7.4}$$

We now present Howe's result.

Theorem 7.2. *[20, Theorem 1.1] There is a constant $C \leq 1/12 + 5\sqrt{2}/6 \approx 1.262$ such that the following statement is true: Given a prime power q, let r be the multiplicative arithmetic function such that for all primes l and positive integers a*

$$r(l^a) = \begin{cases} \dfrac{1}{l^{a-1}(l-1)}, & \text{if } q \not\equiv 1 \pmod{l^c}; \\ \\ \dfrac{l^{b+1}+l^b-1}{l^{a+b-1}(l^2-1)}, & \text{if } q \equiv 1 \pmod{l^c}, \end{cases}$$

where $b = \lfloor a/2 \rfloor$ and $c = \lceil a/2 \rceil$. Then for all positive integers N one has

$$\left| \frac{\#' V(\mathbb{F}_q; N)}{q} - r(N) \right| \leq \frac{CN\rho(N)2^{\nu(N)}}{\sqrt{q}}, \tag{7.5}$$

where $\rho(N) = \prod_{p|N}((p+1)/(p-1))$ and $\nu(N)$ denotes the number of distinct prime divisors of N.

Equipped with Theorem 7.2, we now present and prove our main theorem.

Theorem 7.3. *Let $p \in \{2,3\}$ and let $T_{p^n}(k)$ be defined as above. Then*

(i) *For $3 \leq k < n/4$ we have $T_{2^n}(k) = 2^{n-k+2} + O(2^{k+n/2})$,*
(ii) *For $2 \leq k < n/4$ we have $T_{3^n}(k) = 3^{n-k+1} + O(3^{k+n/2})$,*
(iii) *$T_{2^n} = 3 \cdot 2^n + O(n \cdot 2^{3n/4})$,*
(iv) *$T_{3^n} = 3^{n+1}/2 + O(n \cdot 3^{3n/4})$,*
(v) *$\lim_{n\to\infty} T_{p^n}/(p^n-1) = \begin{cases} 3 & \text{if } p = 2, \\ 3/2 & \text{if } p = 3. \end{cases}$*

Furthermore, in $(i)-(iv)$ the implied constants in the O-notation are absolute and effectively computable.

Proof. By equations (7.3) and (7.4), and Theorem 7.2 with $l = p$, for $3 \leq k \leq n$ we have

$$\left| \frac{T_{2^n}(k)}{2^{n+1}} - \frac{1}{2^{k-1}} \right| \leq \frac{C \cdot 2^k \cdot 3 \cdot 2}{2^{n/2}},$$

from which (i) follows immediately. Similarly for $2 \leq k \leq n$ we have

$$\left| \frac{T_{3^n}(k)}{2 \cdot 3^n} - \frac{1}{3^{k-1} \cdot 2} \right| \leq \frac{C \cdot 3^k \cdot (4/2) \cdot 2}{3^{n/2}},$$

from which (ii) follows. For (iii) we write equation (7.2) as follows:

$$T_{2^n} = \sum_{k=1}^{n} T_{2^n}(k) = \sum_{k=1}^{\lfloor n/4 \rfloor - 1} T_{2^n}(k) + \sum_{k=\lfloor n/4 \rfloor}^{n} T_{2^n}(k).$$

Freely applying (i), the first of the these two sums equals

$$\begin{aligned} & 2^n + (2^n + 2^{n-1} + \cdots + 2^{n-\lfloor n/4 \rfloor + 2}) + O(2^{n/2+2} + 2^{n/2+3} + \cdots + 2^{n/2+\lfloor n/4 \rfloor}) \\ = \quad & 2^n + 2^{n+1} - 2^{n-\lfloor n/4 \rfloor + 2} + O(2^{n/2+\lfloor n/4 \rfloor + 1}) \\ = \quad & 2^n + 2^{n+1} + O(2^{3n/4}) = 3 \cdot 2^n + O(2^{3n/4}). \end{aligned}$$

For the second sum, observe that $p^{k+1} \mid t \implies p^k \mid t$ and so $T_{2^n}(k+1) \leq T_{2^n}(k)$, which gives

$$\sum_{k=\lfloor n/4 \rfloor}^{n} T_{2^n}(k) \leq (3n/4+2) \cdot T_{2^n}(\lfloor n/4 \rfloor) = O(n \cdot 2^{3n/4}).$$

Combining these two sums one obtains (iii). Part (iv) follows *mutatis mutandis*, which together with (iii) proves (v). □

Theorem 7.3 proves that for $k < n/4$, the distribution of the height function $\log_p(|S_p(E_{p^n}(a))|)$ over $a \in \mathbb{F}_{p^n}^{\times}$ is approximately geometric. Hence using an argument similar to the above one can prove that asymptotically, the variance is 2 for $p = 2$, and $3/4$ for $p = 3$. Our proof also gives an upper bound on the number of Kloosterman zeros. In particular, parts (i) and (ii) imply that for $k < n/4$, for increasing k, $T_{p^n}(k)$ is decreasing, and hence the number of Kloosterman zeros is $O(p^{3n/4})$. Shparlinski has remarked [39] that this upper bound follows from a result of Niederreiter [35], which refines an earlier result due to Katz [22]. The Weil bound intrinsic to Howe's estimate fails to give any tighter bounds on $|T_{p^n}(k)|$ for $n/4 \leq k \leq n/2$. Finding improved bounds on $|T_{p^n}(k)|$ for k in this interval is an interesting problem, since they would immediately give a better upper bound on the number of Kloosterman zeros.

While our proof only required the $l = p$ part of Howe's result (when we could have used tighter bounds arising from an Igusa curve argument), the more general form, when combined with our approach, allows one to compute the expected height of the Sylow l-subgroup trees for $l \neq p$ as well, should this be of interest.

8. Test Efficiency

We now address the expected efficiency of Algorithms 2 and 3 when applied to random elements of $\mathbb{F}_{2^n}^{\times}$ and $\mathbb{F}_{3^n}^{\times}$ respectively. Since the number of Kloosterman zeros is $O(p^{3n/4})$, by choosing random $a \in \mathbb{F}_{p^n}^{\times}$ and applying our algorithms, one only has an exponentially small probability of finding a zero. Hence we focus on those n for which such computations are currently practical and do not consider the asymptotic complexity of operations. For comparative purposes we first recall Lisoněk's randomised Kloosterman zero test [28].

8.1. **Lisoněk's Kloosterman zero test.**

For a given $a \in \mathbb{F}_{p^n}^{\times}$, Lisoněk's test consists of taking a random point $P \in E_{p^n}(a)$, and computing $[p^n]P$ to see if it is the identity element $\mathcal{O} \in E_{p^n}(a)$. If it is not, then by Lemmas 2.1 and 2.2 one has certified that the group order is not p^n and thus a is not a Kloosterman zero. If $[p^n]P = \mathcal{O}$ and $[p^{n-1}]P \neq \mathcal{O}$, then $\langle P \rangle = E_{p^n}(a)$ and a is a Kloosterman zero. In this case the probability that a randomly chosen point on the curve is a generator is $1/2$ and $2/3$ for $p = 2$ and $p = 3$ respectively. The test thus requires $O(n)$ point-doublings/triplings in $E_{2^n}(a)$ and $E_{3^n}(a)$ respectively.

8.2. **Algorithm 2 for $E_{2^n}(a)$.**

By Theorem 7.3(v), only one loop iteration of Algorithm 2 is required on average. Each such iteration requires computing: a trace; solving (4.1); a multiplication; a square root; two additions; and a bit-flip. This process has been extensively studied and optimised for point-halving in characteristic 2 [11]. In particular, for $n = 163$ and $n = 233$, point-halving is reported to be over twice as fast as point-doubling [11, Table 3]. Hence in this range of n, with a state-of-the-art implementation, Algorithm 2 is expected to be $\approx 2n$ times faster than Lisoněk's algorithm (or $\approx n$ times faster if for the latter one checks whether or not $\mathrm{Tr}(a) = 0$ before initiating the point multiplication).

For the field $\mathbb{F}_{2^{75}} = \mathbb{F}_2[t]/(t^{75}+t^6+t^3+t+1)$, using a basic MAGMA V2.16-12 [4] implementation of Algorithm 2, we found the Kloosterman zero:

$$\begin{aligned} a &= t^{74} + t^{73} + t^{68} + t^{67} + t^{66} + t^{65} + t^{63} + t^{62} + t^{59} + t^{57} + t^{56} + t^{55} + t^{52} \\ &+ t^{44} + t^{43} + t^{41} + t^{40} + t^{39} + t^{38} + t^{37} + t^{36} + t^{35} + t^{34} + t^{31} + t^{30} + t^{29} \\ &+ t^{28} + t^{25} + t^{24} + t^{23} + t^{22} + t^{19} + t^{16} + t^{15} + t^{14} + t^{13} + t^{12} + t^{11} + t^{8} \\ &+ t^{7} + t^{6} + t^{5} + t^{4} + t^{3} + t^{2} + t, \end{aligned}$$

in 18 hours using eight AMD Opteron 6128 processors each running at 2.0 GHz. Due to MAGMA being general-purpose, without a built-in function for point-halving, the above implementation has comparable efficiency to a full point multiplication by 2^{75} on $E_{p^n}(a)$, i.e., Lisoněk's algorithm. However, using a dedicated implementation as in [11] for both point-doubling and point-halving, one would expect Algorithm 2 to be more than 150 times faster than Lisoněk's algorithm (or more than 75 times faster with an initial trace check). Since point-doubling for the dedicated implementation is naturally much faster than MAGMA's, the above time could be reduced significantly, and Kloosterman zeros for larger fields could also be found, if required.

The $O(n)$ factor speedup is due to the fundamental difference between Lisoněk's algorithm and our approach; while Lisoněk's algorithm traverses the hypothetically-of-order-p^n Sylow p-subgroup tree from leaf to root, we instead calculate its exact height from root to leaf, which on average is 3 and thus requires an expected single point-halving.

8.3. **Algorithm 3 for $E_{3^n}(a)$.** Due to the presence of inversions and square-root computations, one expects each loop iteration of Algorithm 3 to be slower than each loop iteration of Algorithm 2. Indeed our basic MAGMA implementation of Algorithm 3 for curves defined over $\mathbb{F}_{3^{47}}$ runs ≈ 3.5 times slower than our one for Algorithm 2 for curves defined over $\mathbb{F}_{2^{75}}$. However the MAGMA implementation is ≈ 15 times faster than Lisoněk's algorithm in this case (or equivalently 5 times faster if a trace check is first performed).

For the field $\mathbb{F}_{3^{47}} = \mathbb{F}_3[t]/(t^{47}-t^4-t^2-t+1)$, using our MAGMA implementation of Algorithm 3, we found the Kloosterman zero:

$$\begin{aligned} a &= t^{46} + t^{45} - t^{44} - t^{42} + t^{39} - t^{38} - t^{36} - t^{35} - t^{33} - t^{31} - t^{30} + t^{29} + t^{28} \\ &+ t^{26} + t^{25} - t^{24} - t^{22} - t^{21} + t^{20} - t^{19} - t^{17} + t^{16} - t^{15} + t^{14} + t^{13} - t^{11} \\ &+ t^{10} - t^{9} - t^{7} + t^{6} + t^{5} + t^{4} - t^{2} + 1, \end{aligned}$$

in 126 hours, again using eight AMD Opteron 6128 processors running at 2.0 GHz.

In order to improve our basic approach, representational, algorithmic and implementation optimisations need to be researched. It may be possible for instance to improve the underlying point-thirding algorithm by using alternative representations of the curve, or the points, or both. For example, one may instead use the Hessian form [9] of $E_{3^n}(a)$:

$$H_{3^n}(\bar{a}) : \bar{x}^3 + \bar{y}^3 + 1 = \bar{a}\bar{x}\bar{y},$$

where $\bar{a} = a^{-1/3}$, $\bar{x} = -a^{1/3}(x+y)$ and $\bar{y} = a^{1/3}(y-x)$, and an associated tripling formula, see for example [19, §3]. Could point-thirding in this form be faster than that described for the Weierstrass form in Algorithm 3? Also, is there an analogue of the λ-representation of a point [24,38] that permits more efficient point-tripling, and hence point-thirding? We leave as an interesting practical problem the development of efficient point-thirding algorithms and implementations for ternary field elliptic curves with non-zero j-invariant.

9. Concluding remarks

We have presented an efficient deterministic algorithm which tests whether or not an element of $\mathbb{F}_{2^n}^{\times}$ or $\mathbb{F}_{3^n}^{\times}$ is a Kloosterman zero, and have rigorously analysed its expected runtime. Our analysis also gives an upper bound on the number of Kloosterman zeros. By repeatedly applying our algorithm on random field elements, we obtain the fastest probabilistic method to date for finding Kloosterman zeros, which for $\mathbb{F}_{2^n}$ is $O(n)$ times faster than the previous best method, for n in the practical range. Since this method of finding a Kloosterman zero is still exponential in n, it remains an important open problem to compute Kloosterman zeros efficiently.

Acknowledgements

The authors wish to thank Faruk Göloğlu and Alexey Zaytsev for useful discussions, and the reviewers for their comments.

References

[1] Omran Ahmadi and Alfred Menezes. On the number of trace-one elements in polynomial bases for $\mathbb{F}_{2^n}$. *Des. Codes Cryptogr.*, 37(3):493–507, 2005.

[2] K. G. Beauchamp. *Walsh functions and their applications.* Academic Press [Harcourt Brace Jovanovich Publishers], London, 1975. Techniques of Physics, No. 3.

[3] I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography*, volume 265 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, Cambridge, 2000. Reprint of the 1999 original.

[4] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[5] Wouter Castryck and Hendrik Hubrechts. The distribution of the number of points modulo an integer on elliptic curves over finite fields. Preprint, 2011.

[6] Pascale Charpin and Guang Gong. Hyperbent functions, Kloosterman sums, and Dickson polynomials. *IEEE Trans. Inform. Theory*, 54(9):4230–4238, 2008.

[7] Pascale Charpin, Tor Helleseth, and Victor Zinoviev. The divisibility modulo 24 of Kloosterman sums on GF(2^m), m odd. *J. Combin. Theory Ser. A*, 114(2):322–338, 2007.

[8] Pascale Charpin, Tor Helleseth, and Victor Zinoviev. Propagation characteristics of $x \mapsto x^{-1}$ and kloosterman sums. *Finite Fields and Their Applications*, 13(2):366 – 381, 2007.

[9] D.V Chudnovsky and G.V Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *Advances in Applied Mathematics*, 7(4):385 – 434, 1986.

[10] John. F. Dillon. *Elementary Hadamard Difference Sets.* PhD Thesis. University of Maryland, 1974.

[11] Kenny Fong, Darrel Hankerson, Julio Lopez, and Alfred Menezes. Field inversion and point halving revisited. *IEEE Transactions on Computers*, 53:1047–1059, 2003.

[12] Kseniya Garaschuk and Petr Lisoněk. On binary Kloosterman sums divisible by 3. *Des. Codes Cryptogr.*, 49(1-3):347–357, 2008.

[13] F. Göloğlu, P. Lisoněk, G. McGuire, and R. Moloney. Binary kloosterman sums modulo 256 and coefficients of the characteristic polynomial. *Information Theory, IEEE Transactions on*, 58(4):2516–2523, 2012.

[14] Faruk Gölogłu, Gary McGuire, and Richard Moloney. Ternary kloosterman sums modulo 18 using stickelberger's theorem. In Claude Carlet and Alexander Pott, editors, *SETA*, volume 6338 of *Lecture Notes in Computer Science*, pages 196–203. Springer, 2010.

[15] Faruk Göloğlu, Gary McGuire, and Richard Moloney. Binary Kloosterman sums using Stickelberger's theorem and the Gross-Koblitz formula. *Acta Arith.*, 148(3):269–279, 2011.

[16] Faruk Gölogłu, Gary McGuire, and Richard Moloney. Ternary kloosterman sums using stickelbergers theorem and the gross-koblitz formula. Preprint, 2010.

[17] Tor Helleseth and Alexander Kholosha. Monomial and quadratic bent functions over the finite fields of odd characteristic. *IEEE Transactions on Information Theory*, 52(5):2018–2032, 2006.

[18] Tor Helleseth and Victor Zinoviev. On Z_4-linear Goethals codes and Kloosterman sums. *Des. Codes Cryptogr.*, 17(1-3):269–288, 1999.

[19] Huseyin Hisil, Gary Carter, and Ed Dawson. New formulae for efficient elliptic curve arithmetic. In K. Srinathan, C. Rangan, and Moti Yung, editors, *Progress in Cryptology INDOCRYPT 2007*, volume 4859 of *Lecture Notes in Computer Science*, pages 138–151. Springer Berlin / Heidelberg, 2007.

[20] Everett W. Howe. On the group orders of elliptic curves over finite fields. *Compositio Math.*, 85(2):229–247, 1993.

[21] Jun-ichi Igusa. On the algebraic theory of elliptic modular functions. *J. Math. Soc. Japan*, 20:96–106, 1968.

[22] N. M. Katz. *Gauss sums, Kloosterman sums, and monodromy groups.* Princeton Univ. Press, Princeton, NJ, 1988.

[23] Nicholas Katz and Ron Livné. Sommes de Kloosterman et courbes elliptiques universelles en caractéristiques 2 et 3. *C. R. Acad. Sci. Paris Sér. I Math.*, 309(11):723–726, 1989.

[24] Erik Woodward Knudsen. Elliptic scalar multiplication using point halving. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '99, pages 135–149, London, UK, 1999. Springer-Verlag.

[25] K.P. Kononen, M.J. Rinta-aho, and K.O. Väänänen. On integer values of kloosterman sums. *Information Theory, IEEE Transactions on*, 56(8):4011 –4013, aug. 2010.

[26] Gilles Lachaud and Jacques Wolfmann. The weights of the orthogonals of the extended quadratic binary Goppa codes. *IEEE Trans. Inform. Theory*, 36(3):686–692, 1990.

[27] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126(3):649–673, 1987.

[28] Petr Lisoněk. On the connection between Kloosterman sums and elliptic curves. In *Sequences and their applications—SETA 2008*, volume 5203 of *Lecture Notes in Comput. Sci.*, pages 182–187. Springer, Berlin, 2008.

[29] Petr Lisoněk and Marko Moisio. On zeros of kloosterman sums. *Designs, Codes and Cryptography*, 59:223–230, 2011. 10.1007/s10623-010-9457-x.

[30] J. Miret, R. Moreno, A. Rio, and M. Valls. Determining the 2-Sylow subgroup of an elliptic curve over a finite field. *Math. Comp.*, 74(249):411–427 (electronic), 2005.

[31] J. Miret, R. Moreno, A. Rio, and M. Valls. Computing the l-power torsion of an elliptic curve over a finite field. *Math. Comp.*, 78(267):1767–1786, 2009.

[32] Marko Moisio. Kloosterman sums, elliptic curves, and irreducible polynomials with prescribed trace and norm. *Acta Arith.*, 132(4):329–350, 2008.

[33] Marko Moisio. The divisibility modulo 24 of Kloosterman sums on GF(2^m), m even. *Finite Fields Appl.*, 15(2):174–184, 2009.

[34] Marko Moisio and Kalle Ranto. Kloosterman sum identities and low-weight codewords in a cyclic code with two zeros. *Finite Fields Appl.*, 13(4):922–935, 2007.

[35] Harald Niederreiter. The distribution of values of kloosterman sums. *Archiv der Mathematik*, 56:270–277, 1991. 10.1007/BF01190214.

[36] Amílcar Pacheco. Rational points on Igusa curves and L-functions of symmetric representations. *J. Number Theory*, 58(2):343–360, 1996.

[37] Takakazu Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, 15(4):247–270, 2000.

[38] R. Schroeppel. Elliptic curves: Twice as fast! Presentation at the CRYPTO 2000 Rump Session, 2000.

[39] Igor Shparlinski. On the values of kloosterman sums. *IEEE Transactions on Information Theory*, 55(6):2599–2601, 2009.

[40] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992. Corrected reprint of the 1986 original.

[41] Gerard van der Geer and Marcel van der Vlugt. Kloosterman sums and the p-torsion of certain Jacobians. *Math. Ann.*, 290(3):549–563, 1991.

[42] F. Vercauteren. *Computing zeta functions of curves over finite fields*. PhD Thesis. Katholieke Universiteit Leuven, 2003.

[43] Lawrence C. Washington. *Elliptic curves*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2003. Number theory and cryptography.

Claude Shannon Institute, UCD CASL, University College Dublin, Ireland
E-mail address: omran.ahmadi@ucd.ie

Claude Shannon Institute, Dublin City University, Ireland
Current address: Claude Shannon Institute, UCD CASL, University College Dublin, Ireland
E-mail address: rgranger@computing.dcu.ie